What are the three properties that a Cryptographic Hash Function needs to satisfy? Explain the meaning of these properties in math.

Answer: Three properties that a Cryptographic Hash Function needs to satisfy is: preimage resistant, collision resistant and second preimage resistant.

- Preimage resistant
  For the cryptographic hash function:
  $$y = h(x) \qquad (1)$$
  Given the output y of n bits, finding preimages x such that $y = h(x)$ requires $O(2^n)$ time. It should be hard to find a message x with a given hash value y.
- Collision resistant
  For the hash function $h(x)$, it is infeasible to find two distinct values x and x' such that:
  $$h(x) = h(x') \qquad (2)$$
- Second preimage resistant
  Given one message x it should be hard to find another message x' such that:
  $$h(x) = h(x') \qquad (3)$$
  Given an output of n bits and a message x, it requires $O(2^n)$ time before one can find a second preimage x'.

(True or False) In digital signatures, the hash value of the message is encrypted with a user's public key. Explanation is needed.

Answer: The has value of the message should be encrypted with a user's private key rather than the public key. So the answer is false.

Review RSA and Diffe-Hellman algorithms, and then answer the question. Which of the following algorithms can be used for digital signature: RSA, Elliptic Curve, Diffie-Hellman? Explanation is NOT needed.

Answer: RSA, Elliptic Curve.

In ECDSA, the private key is an unpredictably chosen number between 1 and the order of the group. The public key is derived from the private key by scalar multiplication of the base point a number of times equal to the value of the private key.

Answer: True

Prove the theorem regarding the birthday paradox in Lecture 1, slide 16.

Answer:

Theorem:

Let $r_1, r_2, \dots r_n \in 1,2,\dots,N$ be independent, identically distributed integers, then:

$$Pr[r_i = r_j | i \neq j] \geq \frac{1}{2} \text{ for } n = 1.2 \times \sqrt{N} \qquad (4)$$

Proof:

$$\begin{aligned}
Pr[r_i = r_j | i \neq j] &= 1 - Pr(r_i \neq r_j | i \neq j) \\
&= 1 - \prod_{i=1}^{n-1}\left(1 - \frac{i}{N}\right) \\
\text{According to Bernoulli's inequality} \quad & e^x \geq 1 + x: \\
&\geq 1 - \prod_{i=1}^{n-1} e\,xp\left(-\frac{i}{N}\right) \\
&= 1 - exp\left(-\sum_{i=1}^{n-1}\frac{i}{N}\right) \qquad (5) \\
&= 1 - exp\left(-\frac{1}{N}\frac{n^2 - n}{2}\right) \\
&\approx 1 - exp\left(-\frac{1}{N}\frac{n^2}{2}\right) \\
&\text{Then for } n = 1.2 \times \sqrt{N}: \\
&= 1 - exp(-0.72) \\
&\geq \frac{1}{2}
\end{aligned}$$

Recall that in Bitcoin, a user needs to provide a signature in his current transaction in order to spend his UTXO. Whenever a node validates a transaction, it checks the signature on exactly what was signed and rejects the transaction if the signature is invalid. For each transaction signing method listed below, decide if an attacker can steal funds from an input address of a transaction submitted to the Bitcoin network. Explanation is needed.

Section 1

The private key is used to sign the entire transaction (minus the signature).

Answer: The attacker cannot steal funds from the input address of the transaction submitted to the Bitcoin network if the private key is used to sign the entire transaction. Because the private key not only contains the private key information, but also encrypts the transaction together with the private key into the signature. In this way, even if the attacker obtains the signature, the attacker cannot decrypt the private key or modify the transaction information or steal funds.

Section 2

The private key is used to sign the entire output of the transaction and nothing else.

Answer: The attacker can steal funds from the input address of the transaction submitted to the Bitcoin network if the private key is used to sign the output of the transaction but nothing else. Because the signature must include the entire transaction, not just the output, to prove that the user has the ownership of the funds and that the transaction has not been tampered with. The attacker can copy the transaction and change the ID of the transaction in order to done an unauthorized transaction from user.

What does it mean when we say in Bitcoin a transaction is unconfirmed until it has n confirmations?

Answer: According to the longest chain rule, it can not be confirmed that which chain is going to be the longest chain confirmed after n-1 blocks created after this block on the main chain.

Why is it risky for the seller to accept a Bitcoin transaction with 0 confirmation?

Answer: Because the seller is facing the double spending issue, such as Finney attack by a malicious miner. The buyer can includes a transaction sending some coins to himself in his mined block, then the buyer can withholds the mined block and instead send the same coins to a merchant. Once he finds the block and he will broadcasts the block when the merchant accepts the payment and irreversibly provides the service.

Assume each block is mined in 10 minutes, a block has size 1M bytes, and each transaction has an average size of 250 bytes. What is the transaction per second (TPS) Bitcoin network can handle?

$$TPS = \frac{1M}{250}/(10 \times 60) \approx 6.6667 \qquad (6)$$

Scalability of the blockchain is currently a concern. To solve this issue, we can think about increasing the block size or shortening the block generation interval. What are their limitations in terms of increasing TPS?

Answer:

Increasing the block size:

By increasing the block size, more transactions can be stored in a single block, which increases the TPS. However, increasing the block size also increases the size of the blockchain, which can cause that larger blocks can take longer to propagate through the network, which can slow down the overall performance of the blockchain. Also, the probability of forking increases, which will obstruct in increasing the TPS.

Shortening the block generation interval:

By shortening the block generation interval, more blocks can be generated in a given time, which also increases the TPS. However, shortening the block generation interval also increases the difficulty of mining blocks and the proof-of-work difficulty, which can result in centralization of the network. Also, the probability of forking increases, which will obstruct in increasing the TPS.

GASLIMIT is the actual amount of gas spent at the completion of the Block creation.

False

The Ethereum Accounts can send transactions for ether transfer or they can send transactions to invoke a smart contract code. True

Ethereum full node hosts the software needed for transaction initiation, validation, mining, block creation, and smart contract execution.
True

Miner nodes receive, verify, gather and execute transactions.True

A Smart Contract is a piece of code deployed in the Blockchain node. Execution of a smart contract is initiated by a message embedded in a transaction.
True

The attacker has a branch with only m additional blocks, and both are trying to extend their respective branches. Assume them honest network and the attacker has a proportion of p and q of the total network hash power, respectively.

Section 1

Let a_z denote the probability that the attacker will be able to catch up when he is currently z blocks behind. Find out the closed form for a_z with respect to p, q and z. Detailed analysis is needed. (Hint: a_z satisfies the recurrence relation $a_z = pa_z + 1 + qa_z - 1$.)

Answer: According to the hint, a_z satisfies the recurrence relation $a_z = pa_{z+1} + qa_{a-1}$, reform the equation in terms of $a_z - a_{z-1}$:

$$\begin{aligned} a_z - a_{z-1} &= \left(\frac{q}{p}\right)^{z-1}(a_1 - a_0) + a_0 \\ a_z &= \sum_{i=0}^{z-1}\left(\frac{q}{p}\right)^{z-1}(a_1 - a_0) + a_0 \end{aligned} \quad (1)$$

Where the a_0 should be 1 in this case since the attacker has the same block as the chain (p=q):

$$\begin{aligned} a_z &= \sum_{i=0}^{z-1}\left(\frac{q}{p}\right)^{z-1}(a_1 - a_0) \\ &= \sum_{i=0}^{z-1}\left(\frac{q}{p}\right)^{z-1}(a_1 - 1) + 1 \\ &= (a_1 - 1)\frac{1-\left(\frac{q}{p}\right)^z}{1-\frac{q}{p}} + 1 \\ &= a_1 z - k + 1 \end{aligned}$$

Assume $z \to \infty$, $a_z = 0$: (2)

$$\begin{aligned} \lim_{z\to\infty}(a_1 - 1)\frac{1-\left(\frac{q}{p}\right)^z}{1-\frac{q}{p}} + 1 &= 0 \\ a_1 &= 1 - \frac{1-\frac{q}{p}}{1-\left(\frac{q}{p}\right)^z} \\ a_z &= \begin{cases} \left(\frac{q}{p}\right)^z, & \text{p > q} \\ 1, & \text{otherwise} \end{cases} \end{aligned}$$

Section 2

Compared with the Bitcoin white paper, we model m more accurately as a negative binomial variable. m is the number of successes (blocks found by the attacker) before n failures (blocks found by the honest network), with a probability q of success. Show that the probability for a given value m is $P(m) = \big((m+n-1)Cm\big)p^n q^m$.

Answer: Since that the equation of the probability of an event with m time successes and n times of failures can be written as:

$$P = q^m p^n \quad (3)$$

In the trail, the last trail must be failure so the last successes is coming before the last failure, so there should be total $m + n + 1$ trails. Hence, it can be written as:

$$P(m) = \binom{m+n+1}{m} q^m p^n \quad (4)$$

Suppose a mining pool wants to blacklist transactions from address X. In other words, they want to freeze the money held by that address, making it unspendable.

Section 1

The mining pool announces that they will refuse to work on a chain containing a transaction originating from address X. Explain why this strategy can guarantee that the blacklisted transactions will never be published if the mining pool has the majority of the hash power.

Answer: If a transaction from address X is included from the miner, then the pool will fork since the pool has the most of the hash power, which will create a longer chain in order to invalidates the chains contains the transaction from the address X. Hence, the miner will be informed that this block will be invalidated.

Section 2

The mining pool announces that they will attempt to fork if they see a block that has a transaction from address X, but they will give up after the transaction from address X has k confirmations. The success of this attack depends entirely on the motivation of other miners to join the attacker. If a miner includes a transaction from address X in his block, he will receive block reward plus transaction fee from address X. Otherwise, the miner only receives block reward. Suppose the attacker controls q = 20% of the network hash power. Let k = 2 and block reward be 12.5 BTC ≈$48, 550. What is the minimum transaction fee address X has to pay in order to avoid being blacklisted?

Answer: The probability that the attacker successfully build the block is q^2. Hence, in order to avoid attacking, the payment amount should be:

$$(1 - q^2)(\text{transaction fee} + \text{reward}) \quad (5)$$

Where the total amount should be greater than the block reward, hence:

$$\begin{aligned} (1 - q^2)(\text{transaction fee} + \text{reward}) &\geq \text{reward} \\ \text{transaction fee} &\geq \frac{\text{reward}}{1 - 0.2^2} \\ \text{transaction fee} &\geq \frac{12.5}{0.96} - 12.5 \\ \text{transaction fee} &\geq 0.5208 \text{ BTC} \end{aligned} \quad (6)$$

Describe all events that cause each transition in the state transition diagram in Lecture 3, Slide 35

Begin	End	Probability	Event
0’	0	$(1-\gamma)(1-\alpha)$	Honest miner mines a block on the main branch
0’	0	$\gamma(1-\alpha)$	Honest miner mines a block on the selfish miner’s block
0’	0	α	Selfish miner mines a block on the private branch
0	0	$1-\alpha$	Honest miner mines a block
1	0’	$1-\alpha$	Honest miner mines a block on the main branch, selfish miner publishes the private branch containing one block
2	0	$1-\alpha$	Honest miner mines a block on the main branch, selfish miner publishes the private branch containing two blocks
n	n+1	α	Selfish miner mines a block on the private branch
n	n-1	$1-\alpha$	Honest miner mines a block on the main branch

At which transitions in the state transition diagram in Lecture 3, Slide 35, the honest miners would earn the block reward, and how many?

Begin	End	Probability	Reward
0’	0	$(1-\gamma)(1-\alpha)$	2
0’	0	$\gamma(1-\alpha)$	1
0	0	$1-\alpha$	1